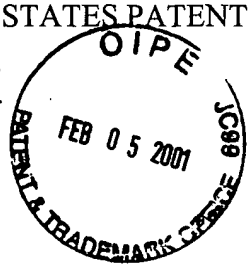
P20229

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE



Applicant : Pek Yew TAN et al.

Serial No. : 09/709,726 Group Art Unit

Filed : November 13, 2000 Examiner

For : TIME BASED MULTIMEDIA OBJECTS STREAMING APPARATUS AND METHOD

CLAIM OF PRIORITY

Commissioner of Patents and Trademarks
Washington, D.C. 20231

Sir:

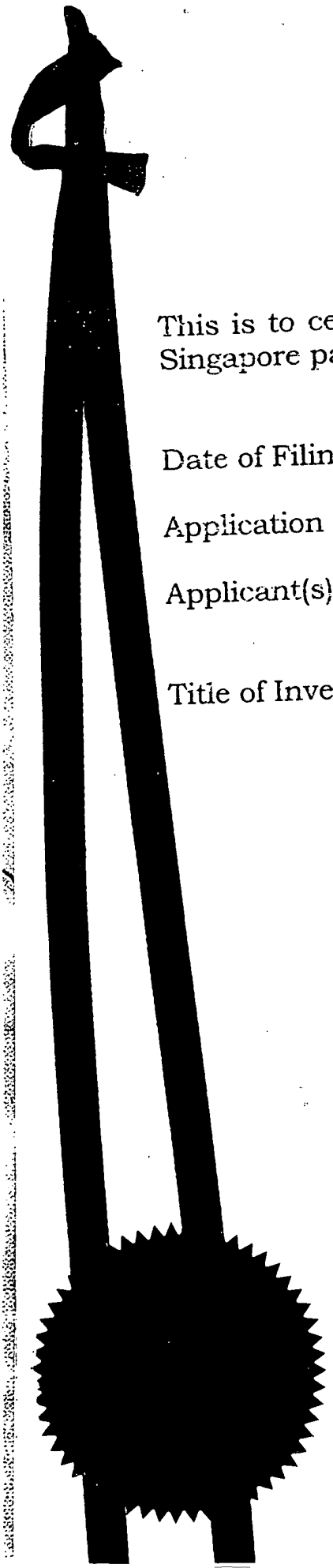
Applicant hereby claims the right of priority granted pursuant to 35 U.S.C. 119 based upon Singapore Application No. 200000085-1, filed January 07, 2000. As required by 37 C.F.R. 1.55, a certified copy of the Singapore application is being submitted herewith.

Respectfully submitted,
Pek Yew TAN et al.

Bruce H. Bernstein
Reg. No. 29,027

February 5, 2001
GREENBLUM & BERNSTEIN, P.L.C.
1941 Roland Clarke Place
Reston, VA 20191
(703) 716-1191

P. TAN e191
09/779,726
NOV 06 2001 TECH CENTER 2600
PATENT & TRADEMARK OFFICE JC99
FEB 05 2001



REGISTRY OF PATENTS
SINGAPORE

This is to certify that the annexed is a true copy of the following Singapore patent application as filed in this Registry.

Date of Filing	:	07 JANUARY 2000
Application number	:	200000085-1
Applicant(s)	:	MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD
Title of Invention	:	TIME BASED MULTIMEDIA OBJECTS STREAMING APPARATUS AND METHOD

CERTIFIED COPY OF
PRIORITY DOCUMENT

Tan Kar Leng(Miss)
Assistant Registrar
for REGISTRAR OF PATENTS
SINGAPORE

26TH MAY 2000

2000000085-1

PATENTS FORM 1

SINGAPORE
THE PATENTS ACT
(CHAPTER 221)
PATENTS RULES

Rule 19

The Registrar of Patents
Registry of Patents

REQUEST FOR THE GRANT OF A PATENT

THE GRANT OF A PATENT IS REQUESTED BY THE UNDERSIGNED ON THE BASIS OF THE PRESENT APPLICATION.

I. Title of Invention	TIME BASED MULTIMEDIA OBJECTS STREAMING APPARATUS AND METHOD	
II. Applicant (s) *(See note 2)*	(a) Name	Matsushita Electric Industrial Co., Ltd
	Body Description / Residency	A company organized and existing under the laws of Japan
	Street Name & Number	1006, Oaza Kadoma, Kadoma-shi,
	City	Osaka 571-8501
	State	
	Country	Japan
	(b) Name	
	Body Description / Residency	
	Street Name & Number	
	City	
	State	
	Country	
	(c) Name	
	Body Description / Residency	
	Street Name & Number	
	City	
	State	
	Country	

200000085-1

III. Declaration of Priority (see note 3)	Country / Country Designated		File No.	
	Filing Date			
	Country / Country Designated		File No.	
	Filing Date			
	Country / Country Designated		File No.	
	Filing Date			
	Country / Country Designated		File No.	
	Filing Date			

IV. Inventors (*See note 4*) (a) The applicant(s) is/are the sole/joint inventor(s).	☐ YES ☒ NO
(b) A statement on Patents Form 8 is/will be furnished.	☒ YES ☐ NO
V. Name of Agent (if any) (*See note 5*)	**LLOYD WISE**

VI. Address for Service (*See note 6*)				
	Block/Hse No		Level No	
	Unit No/PO Box	P O BOX 636	Postal Code	910816
	Street Name			
	Building Name	TANJONG PAGAR POST OFFICE		

VII. Claiming an earlier filing date under section 20 (3), 26(6) or 47(4) (*see note 7*)			
	Application No		
	Filing Date		

[Please tick in the relevant space provided]:

() Proceeding under rule 27(1)(a).
Date on which the earlier application was amended = _____

or

() Proceeding under rule 27(1)(b).

VIII. Invention has been displayed at an International Exhibition (*See note 8*)	☐ YES ☒ NO		
IX. Section 114 requirements (*See note 9*)	The invention relates to and/or used a micro-organism deposited for the purposes of disclosure in accordance with Section 114 with a depository authority under the Budapest Treaty ☐ YES ☒ NO		
X. Check List (*To be filled in by applicant or agent*)	A. The application contains the following number of sheet(s) :-		
	1. Request.	4	Sheets
	2. Description.	28	Sheets
	3. Claim (s).	9	Sheets
	4. Drawing (s).	10	Sheets
	5. Abstract.	1	Sheets
	B. The application as filed is accompanied by :-		
	1. Priority document.		
	2. Translation of priority document.		
	3. Statement of Inventorship & right to grant	X	
	4. International Exhibition Certificate.		
XI. Signature (s) (*See note 10*)	Applicant (a)		
	Date	7 January 2000	
	Applicant (b)		
	Date		
	Applicant (c)		
	Date		

NOTES :

1. This form when completed, should be brought or sent to the Registry of Patents together with the prescribed fee and 3 copies of the description of the invention, and of any drawings.

2 Enter the name and address of each applicant in the spaces provided at paragraph II. Names of individuals should be indicated in full and the surname or family name should also underlined. The names of all partners in a firm must be given in full. The place of residence of each individual should also be furnished in the space provided. Bodies corporate should be designated by their corporate name and country of incorporation and, where appropriate, the state of incorporation within that country should be entered where provided. Where more than three applicants are to be named, the names and address of the fourth and any further applicants should be given on a separate sheet attached to this Form together with the signature of each of these further applicants.

3. The declaration of priority at paragraph III should state the date of the previous filing, and the country in which it was made and indicate the file number, if available. When the application relied upon in an International Application or a regional patent application e.g. European patent application, one of the countries designated in that application [being one falling under the Patents (Convention Countries) Order] should be identified and the name of that country should be entered in the space provided.

4. Where the applicant or applicants is/are the sole inventor or the joint inventors, paragraph IV should be completed by marking the "YES' Box in the declaration (a) and the 'NO' Box in the alternative statement (b). Where this is not the case, the 'NO' Box in declaration (a) should be marked and a statement will be required to be filed on Patents Form 8.

5. If the applicant has appointed an agent to act on his behalf, the agent's name should be indicated in the spaces available at paragraphs V.

6. An address for service in Singapore to which all documents may be sent must be stated at paragraph VI. It is recommended that a telephone number be provided if an agent is not appointed.

7. When an application is made by virtue of section 20(3), 26(6) or 47(4), the appropriate section should be identified at paragraph VII and the number of the earlier application or any patent granted thereon identified. Applicants proceeding under section 26(6) should identify which provision in rule 27 they are proceeding under. If the applicants are proceeding under rule 27(1)(a), they should also indicate the date on which the earlier application was amended.

8. Where the applicant wishes an earlier disclosure of the invention by him at an International Exhibition to be disregarded in accordance with section 14(4)(c), then the 'YES' box at paragraph VIII should be marked. Otherwise the 'NO' box should be marked.

9. Where in disclosing the invention the application refers to one or more micro-organisms deposited with a depository authority under the Budapest Treaty, then the 'YES' box at paragraph XI should be marked. Otherwise the 'NO' box should be marked.

10. Attention is drawn to rules 90 and 105 of the Patent Rules. Where there are more than three applicants, see also Note 2 above.

11. Applicants resident in Singapore are reminded that if the Registry of Patents considers that an application contains information the publication of which might be prejudicial to the defence of Singapore or the safety of the public, it may prohibit or restrict its publication or communication. Any person resident in Singapore and wishing to apply for patent protection in other countries must first obtain permission from the Singapore Registry of Patents unless they have already applied for a patent for the same invention in Singapore. In the latter case, no application should be made overseas until at least two months after the application has been filed in Singapore.

For Official Use

Application Filing Date	:	/ /
Request received on	:	/ /
Fee received on	:	/ /
Amount	:	
* Cash / Cheque / Money Order No :		

** Delete whichever is inapplicable.*

Our Ref: SP3717

SP371705.doc

07 JAN 2000

TIME BASED MULTIMEDIA OBJECTS STREAMING APPARATUS AND METHOD

BACKGROUND AND FIELD OF THE INVENTION

This invention relates to the transmission of real time data between a sender and receiver over a transmission network.

The current delivery of real-time data depends very much on the feed back channel from the receiver to the sender which provides a report on the received data transmission. Based on this feedback report, the sender will take appropriate action for the transmission of subsequent data to the receiver. The action taken by the sender may require reservation of more network resources for end-to-end data delivery and re-transmission of lost data. Re-transmission of lost data requires the receiver end to report specifically the amount of lost data encountered before the entire data can be of use to the applications at the receiver end. For a large system having a large number of servers and clients, the report messages between the terminals utilise a considerable amount of bandwidth.

In both connection and connectionless data networks, data packets at the lower layers such as the network layer differentiate data only by priority or time-to-live parameters. The distinction made between data packets at the network layer does not relate the coupling nature of the application layer data fragmented into data packets for network layer delivery. Some of these data packet parameters

used to distinguish the various classes of service between data packets are listed in the IETF (Internet Engineering Task Force) RFC 791 on Internet Protocol version 4 and IETF's RFC 2460 on Internet Protocol version 6. In the current specifications for Differentiated Service as mentioned in IETF's RFC 2474, RFC 2475, RFC 2597 and RFC 2958, only a means to provide data packet marking is provided. IP data packets with special marking allow the network components such as routers or switches to treat the IP data packets based on the packet's different Quality of Service fields or DSCP, as an indicator on how data flow should be determined. The re-transmission and dropping of data packets belonging to a specific stream is performed at the IP packet level without any knowledge of the application data. Data packet differentiation is only performed at the packet level and selective dropping of a group of data packets is almost impossible if this process has to be performed at the network level.

Without the additional information from the application software, a data packet that makes it though the network may consist of only part of the data the application software needs. With the re-transmission of the missing data that finally arrives at the receiver end, a strict real-time requirement in most instances cannot be met. A data object that arrives not adhering to the real-time requirement is said to be of no presentation value because the time for decoding the piece of information has already passed. As a result, network resources are used in delivering real-time data of no value to the end user. In addition, the non-presentation value group of data packets may cause

unnecessary network congestion at the already strained intermediate network nodes. In many instances the congestion of network traffic may be prolonged and in a worst case scenario the traffic congestion may spread to a wider area.

For most real-time data streaming across a packet data network, the decoding rate for the data is only made known to the end devices, for instance the server and the end client. In many instances, the server can only deliver data at the rate based on the decoding rate of the data and a report sent from the client giving statistics of the data received. In some implementations, a large data buffer or an arbitrary size of data buffer needs to be pre-allocated at the client side before the streaming of data from the server to the client can begin. In the scenario where a wide area network is constructed based on a number of intermediate network nodes such as routers and switches which have no information on the type of data and how the data should be treated for end-to-end delivery, the intermediate nodes have no real-time streaming knowledge of the network packets arriving and leaving the nodes. In the event of network congestion at one of the intermediate nodes, there is no means to deliver or re-schedule the delivery of network data packets that makes sense from the view of the application data. In most cases, only a partial object data is successfully delivered to the end terminals and re-transmission of lost data may be performed automatically by the network layer.

SUMMARY OF THE INVENTION

According to the invention in a first aspect, there is provided a method of sending a data object through a communications network from a sender to a recipient via at least one network node comprising the steps of:

a. dividing the data object into a stream of data packets to be transmitted,

b. marking each data packet with a delivery time; and

c. discarding a said data packet en route when the delivery time of the said packet cannot be met.

According to the invention in a second aspect, there is provided a method of creating synchronised time stamped multimedia objects comprising the steps of:

a. fragmenting the data objects into one or more data packets;

b. marking each data packet containing a part or full portion of a single multimedia object using a unique reference;

c. creating a link for related fragmented network data packets originating from the same multimedia object using an object reference before transmission;

d. stamping the data packets related to the object reference with a time reference marking the instance the data packet leaves a transmission node; and

e. attaching a precedence level of the object based on the object reference.

According to the invention in a third aspect, there is provided a method for scheduling the transmission time of multimedia objects comprising the steps of:

a. dividing the transmission time into frames based on an object decoding rate;

b. further dividing the transmission time within each frame of the multimedia object channel into time slots, the number of time slots being equivalent to dividing the transmission rate by the multimedia object decoding time;

d. forming a delivery time schedule of multimedia objects with a time stamp within the scheduled object delivery period or frame period;

e. computing a scheduled object delivery period which is equivalent to the maximum object group delay period allowable; and

f. re-scheduling objects that cannot be delivered within a current time slot to a later time slot.

According to the invention in a fourth aspect, there is provided apparatus for interconnecting an object streaming apparatus into existing network interconnection apparatus to control data traffic generated by the multimedia object streams comprising:

a. an ingress gateway to control entry of data packets into a network;

b. a content traffic exchange hop attached to legacy switching or a routing network component; and

c. an egress gateway to control the exit of data packets from the network and to stream content to end-users.

The described embodiment of the invention uses a system architecture that allows end-to-end streaming of time based objects. An input data object from

an object stream entering the network is marked and disassembled by an ingress object traffic controller.

In the ingress object traffic controller, a large data object is fragmented into a plurality of smaller network data packets as specified by the chosen network layer for delivery. Each network data packet is marked according to the precedence level desired by the application entities at the source of the object stream. To provide the network data packets with linkage and real-time information, each object has an object descriptor created. Based on the real-time requirement and the network resources allocated for the delivery of the object stream, a time for an object frame is computed and created for the delivery of time-stamped object data. The data information of the object descriptor is mapped to the header of the network data packet where possible or is encoded into the payload.

At each of the intermediate nodes such as routers and switches connected with the apparatus, the data packet is received. The data packet is re-routed to the host using an egress object traffic controller if the data object has reached the final destination, otherwise the packet is processed by an exchange object traffic controller for re-transmission to another network node. In the egress or exchange object traffic controller, the object descriptor for each stream is collected and a timer circuit is spawned for each object stream. Based on the timing information, a system object frame is created dynamically and the number of time slots per object frame is computed. The time slots are used to

mark the delivery time of the object data. The time schedule for the delivery of the object based on time slots is performed based on a traffic shaper or filter adopted for the stream. Object data meeting the criteria of the traffic shaper is put in the object stream queue and object data not meeting the requirement of the traffic shaper is stored in a cache memory for a generalised lower priority object data scheduling. An object decoding period is determined from the set of object stream descriptors received. One or more object decoding time slots are allotted for the object data to be delivered to meet real-time criteria in the stream object queue. After completing scheduling the object data that will expire within the object time frame in the stream object queue and data objects in the stream object cache memory are scheduled next. After the object has been scheduled, excess time slots are saved for future object frames. If there are insufficient time-slots, additional time-slots are allocated from the reserves to meet the need. If there are insufficient time-slots for delivery, object data that expires in the next object frame is discarded.

Data object packets scheduled for re-transmission are then re-assembled based on the output network criteria and a network header is mapped accordingly. The process is repeated for the data object in the exchange object traffic controllers of the intermediate nodes within the service provider domain until the final destination has been reached.

The described embodiment of the invention provides apparatus and a method for the delivery of scalable multimedia objects to meet real-time applications on

a heterogeneous network. These scalable multimedia objects may contain synchronised audio, video and data information for interactive, communicative and broadcast applications over a wide area network. The described embodiment provides a multimedia objects streaming method based in network resource availability at each network component and it allows real-time end to end delivery of multimedia objects with minimal effect on the presentation quality of the multimedia object over networks with dynamically changing network resource parameters such as varying bandwidth, delay and latency jitters. The described embodiment provides a method and apparatus that allows fast high level application data context switching and routing at the network layer.

The described embodiment of the invention further allows the delivery of data without additional feedback from the receiver and thus cuts down on the complexity in the protocol needed by both ends of the users to handle a feedback report. The processing power needed for the client to perform the reporting based on the data received from the server is also reduced.

The described embodiment further provides a means to deliver real-time and near real-time multimedia object delivery of different decoding rates in a connectionless oriented connection accurately based on individual decoding rate. The described embodiment further allows synchronisation of the time based object streams such that real-time delivery specified for the object

stream can be meet accurately without an accurate timer being implemented by the host network node.

The described embodiment provides a mechanism to perform network congestion avoidance at each intermediate network node dynamically at the rate equivalent to the smallest decoding rate from a set of decoding rates from the object streams and also allows link information between data packets belonging to the same data object to be retained as the data is being fragmented and delivered across the network.

For real time delivery, the intermittent delay is kept at the maximum constant delay such that end to end delay for the arrival of all data objects is kept at a constant rate over time.

The content data switching can be performed at the network level, thus freeing the application level from the rudimentary work of re-transmitting of content object data received.

With the ability to couple fragmented data packets originating from a single data object, the described embodiment enables routers or intermediate network nodes between end-to-end users to drop data packets of the data objects with no presentation value. A data object without presentation value is one where the data object cannot arrive at the end user at the decoding rate. This prevents

non- presentation data objects causing network congestion by dropping the said data objects.

A timer to improve synchronisation without the need for a central control time or network time protocol to keep synchronisation between network nodes is provided. The described embodiment allows an accurate means to estimate the decoding time of each object belonging to different object streams.

The described embodiment also allows a single network traffic controller to control object streams with different traffic shaper or filtering algorithms.

The described embodiment allows the existing network configuration to use the implementing apparatus to control real-time traffic over wide area network such as the internet or corporate wide intranet. The described embodiment also provides an infrastructure for the content provider to stream AV content to the users in real-time and the amount of data objects to be delivered to end used can varies depending on the profile of the users on the data packet network such as the internet. For a virtual leased line, the server can choose to stream content to their valued clients with better quality of service over the non-valued clients.

BRIEF DESCRIPTION OF THE DRAWINGS

An embodiment of the invention will now be described, by way of example, with reference to the accompanying drawings, in which:

FIG. 1 illustrates the management of streamed data traffic from a multimedia server or any other client user terminal in a content provider domain to the receivers at a content consumer via a service provider domain. This figure also shows the interconnections of content stream traffic control for the three domains for the management of network resources for the efficient delivery of content.

Fig. 2a shows a preferred format of an object descriptor and fig. 2b shows a flowchart for creating a object reference from an input object stream.

Fig. 3 is a block diagram of an ingress object traffic controller used in controlling the flow of real-time traffic into the network.

Fig 4 is a block diagram of an egress object traffic controller and an exchange traffic controller used in the implementation of an egress network node and intermediate node for controlling the flow of real-time data.

Fig. 5 is a structure of an object frame and the time slots within each object frame. This also illustrates the way the number of time slots can increase or decrease in accordance with traffic conditions.

Fig. 6 illustrates a preferred implementation of a timer circuit of a generic object traffic controller for the egress and exchange nodes.

Fig. 7 shows a detailed block diagram of object packet stream traffic control found in the object stream scheduler sub-block of the exchange object traffic controller.

Fig.8 shows a detailed block diagram of the time slot allocator found in the object stream scheduler sub-block of the exchange object traffic controller.

Fig. 9 shows a flowchart of the method of scheduling object data in the time slot allocated for the object frame. This figure also shows how time slots not used for the current object frame are accumulated for future use.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The described embodiment of the invention provides a system and apparatus for, and methods of, delivering multimedia objects for improved streaming at the multimedia content provider, improving the quality presentation at content consumer end and improving the network congestion at the service provider domains. The embodiment is described in the following in detail with reference to accompanying drawings. In the following description, numerous details are set forth in order to provide for thorough understanding of the described

embodiment of the present invention by one skilled in the art. In some of the explanations, some obvious details that forms part of the embodiment are left out in order to not unnecessarily obscure the description.

Fig. 1 is a diagram of an exemplary network system consisting of different network functional nodes for implementing the various aspects of the embodiment of the present invention. This figure shows an additional network functional component being added to allow traffic control and management of data generated by the users in two domains: content producer 1.1, and content consumer 1.10. The physical end-users can at different instances be in either the content consumer or content producer domains depending on the flow of the object stream. The entity that ensures the end-to-end delivery of object streams is the service content provider,1.5. The service provider domain 1.5 is connected to the content producer 1.1 via a network link 1.4, that shares the same type of network protocol and architecture as a first connected network switching element,1.6. For brevity, the network component 1.6 is termed a switching element which is a representative network component for routing of data packets or performing label switching or layer 2/3 switching.

The service provider may contain a network of interconnected switching elements that support different network protocols and architecture. Each of the switching elements is connected to a content traffic exchange hop, 1.8, to perform the function of object data interchange and to control the flow of object traffic out the switching element. The connectivity between the switching

element and the content traffic exchange hop 1.8 is via a standard link, 1.7, that allows data to be exchanged. The link 1.7 allows selected data stream flow from different end-to-end users to be controlled and managed by the content traffic exchange hop 1.8 before being passed back to the switching element 1.6.

In the content provider domain 1.1, traffic is generated by each server entity, 1.2, and is moderated by an ingress gateway, 1.3 to control the traffic entering the service provider domain. For brevity, the term "server" is used to describe a logical entity which is the source of content to be delivered across the network. The last switching element 1.6 in the service provider domain is connected to an egress gateway 1.11 of the content consumer domain 1.10, via an interconnect, 1.9, that shares the same type of network protocol and architecture.

The content consumer domain 1.10 may consist of clients with different network access profiles. These clients can either have high bandwidth, 1.14, fixed line access, 1.12, or low bandwidth, 1.15, mobile access, 1.13. The egress gateway 1.11 will regulate the traffic streamed to the client and the requests from the server carried out by the client.

Reference numeral 2.1 of fig 2a shows an exemplary descriptor used by multimedia objects these are sent to the network via an object fragmentor. The object descriptor 2.1 provides sufficient information for the object fragmentor to

enable data packets belonging to the same object to be linked. Data packet linking is necessary as data objects from the application, in many cases if not all cases, need to be fragmented into smaller packets adapted for the network delivery layer before the objects can be delivered by the data carriers of the network layer. The descriptor includes a time value, a major object reference, a minor object reference, a decoding rate for the object, a precedence offset for the object which provides priority of delivery information and the object size.

The flowchart in fig 2b shows the method for providing time information and linking of data packets for a fragmented multimedia object. For each transmission of object data, a session is set-up with the object fragmentor, 2.2. The object fragmentation process is then carried out based on the descriptor's decoding rate and precedence offset parameter at step 2.3. This initialisation process allows the fragmentor to decide on the object marking value of the payload of the network delivery data packet. Based on the object descriptor and the network delivery payload size, the number of data packets to be linked is determined at step 2.4. Each of the data payloads for the fragmented object is then stamped with a unique major object reference number, a number which the object stream uses for the entire life cycle of the session. This major object reference number is incremented by one for each session created by the application with the object fragmentor at step 2.5. A minor object reference number is also created starting at one and incremented for each packet of the object. For each data packet created from the object data, both the major and minor object reference numbers are stamped either as part of the data packet

payload or packet data header or a representative of header information that the network data packet carries. The steps of stamping the data packet and packing of fragmented data into the payload of data packets are performed at steps 2.6, 2.7 and 2.8. When all the network data packets have been marked, the network data packets are time stamped, at step 2.9, with the same time value passed by the object descriptor. The object fragmentor will synchronise with coming object data if the decoding time rate is the same based on the time value in the object descriptor of the object reference.

Fig 3 shows the block diagram of an ingress object traffic controller of the gateway 1.3 enabling marking and controlling the flow of object data entering the network. Object data enters the ingress object traffic controller with the object descriptor, 2.1, passed down from the application entities. The object data is fragmented into smaller data packet payloads by the objector fragmentor, 3.5 based on the input network payload size, 3.2. For each data object divided into smaller data packets, the time is marked by the timer or time value in the object descriptor, 2.1 by the functional block 3.6. The timer derives it's time value from that of the descriptor, 2.1. An input decoding rate and the time value are fed into the timer block, 3.6, via the input 3.3 if the time values from subsequent object descriptors are the same. The timer value generated is fed into time slot divider, 3.8, where time slot transmission of object data time slots are created based on the bandwidth allocated for the delivery medium. Based on the time stamped on the data packets belonging to the same object, a time slot allocator 3.9 determines in which time slot the group of data packets

is to be delivered to the network. For each data packet to be delivered, the data packet is mapped to a delivery network payload packetizer module, 3.11 which translates the timing and the object reference of the object descriptor to the header of the network data packet or encodes this as part of the network data packet's payload and the time information of the data packet is sent to a network parameter mapper module, 3.10 which encapsulates the fragmented object data packet into the format used for delivery to the network layer before the whole network packet is delivered to the network delivery medium, 3.12.

Fig 4 shows a block diagram of apparatus that performs the full functions of the egress object traffic controller of the gateway 1.1 and a content exchange traffic controller of the content traffic exchange hop 1.8. The egress object traffic controller consists of the functional blocks 4.5, 4.6, 4.4, 4.10 and 4.2. For each network data packet received, the data packet header is parsed and decoded by the network parameter decoding function of block 4.5. The timing information and the object reference extracted from the header are passed for further processing to a timer circuit, 4.4.

Another source of timing information and object reference number can be extracted from the payload by the network payload de-packetizer function of functional block 4.6. The timing and object reference are only extracted from the network packet payload if the network header of the network data packets does not carry timing information and object reference number. Encoding of the

timing information and the object reference number in the payload of the data packet follows the object descriptor shown in Fig.2a.

The timer circuit, 4.4, contains a time stamp for each object data packet received within the object decoding frame. A handle reference is created to provide a distinction between the various object data packets received based not only on the object reference number extracted from the object descriptor but also the source and destination network address and the numeral representative of the network service access point. For the timer for the object received to work accurately within each frame, the timers in block 4.4 are updated or created by timing information from an object stream timer circuit, 4.12. The timing information is also passed to a data de-fragmentor, 4.2 for accurate construction of a data stream if the data stream received has reached the final destination. For the object stream received, the stream is further processed by an object stream classifier, 4.10, to determine a final destination of the object stream based on the unique handle reference. If the object stream has reached the final destination, the data packets are passed to the object de-fragmentor, 4.2. In order for the object to be re-assembled and later fed to the buffer used for object decoding, timing information from the timer circuit 4.12 which keeps the timers for the object stream received within one object decoding time frame is required.

In order for the host network components such as switches and routers to perform the function of a content exchange traffic controller, the functional sub-

blocks in the object stream scheduler block, 4.7, are required. The internal clocking of the timer referenced by the unique handle reference providing all the accurate time keeping is kept by the object stream timer circuit, 4.12. This contains timing information for the object stream to be re-routed to another destination. For each stream received, the delivery time slots for each object data packet, based on each individual object stream decoding timing requirement and the transmission time required, are computed by a time slot divider, 4.11. In addition, divider 4.11 also provides an object packet stream traffic control, 4.8 with the object descriptor constructed from the data packet received via the network parameter mapper and decoder functional block, 4.5. Based on all of the information computed for each object stream of the time slot divider, the object stream traffic control, 4.8, then decides the time required for the final delivery of the object stream. Based on the precedence of the object stream carried in the object descriptor of each object stream, the time slot allocator 4.9 performs the final decision as to how the object data fragmented into data packets is delivered out to the network via the functional block 4.6. Timing information of the data object stream that is scheduled to be transmitted is performed by a system time slot divider, 4.3. which gathers all the information of each time slot divider 4.11 assigned to each object stream to determine a minimum object decoding time. The number of time slots for the node is determined by equation eq2 below . An object frame is determined by the system time slot divider in accordance with equation eq.1 below.

Object Frame =

$$\text{Min}(\text{DecodingRate}_{\text{Object Stream 1}}, \text{DecodingRate}_{\text{Object Stream 2}},..,\text{DecodingRate}_{\text{Object Stream N}})$$

..... (EQ.1)

System time slot divider 4.3 performs and maintains a universal object frame for the content exchange traffic controller based on the network resources allocated for transmission. The network resources may include bandwidth, latency and total packet group delay. Information generated by divider 4.3 is also synchronised with the time slot allocator 4.9, such that the same timing information can be mapped to the network data header or encoded in the payload of the network packet carried out by the functional block 4.5.

To enable the exact transmission time of the object data to be transmitted out at the egress node or content traffic exchange hop, a time frame called the object frame is assigned for the object streams to be transmitted out. The pre-allocated time period for the object frame is equivalent to the time for decoding one object data as described in the object descriptor. In the case where there is a requirement to use more or less bandwidth, the object frame period can be increased and reduced. Each object frame may consist of more than one time slot depending on the bandwidth allocated for the delivery of the object data. Depending on the amount of transmission bandwidth allocated, the object frame may contain more than one time slot. The number of time slots per object frame can be computed as follows:-

No. of Time Slots (n) = Transmission rate/Object Decoding rate

(EQ. 2)

For each object frame, only an integer number of time slots is allowed. In the case of a non-integer value as a result of eq. 2, an additional time slot may be allocated for each time object frame if the sum of the fractional portion of time slots calculated forms a value more than one accumulated over a number of object frames. The rate of additional time slots in terms of object frame is inversly proportional to the fractional portion of the computed time slots, n, derived from eq. 1. Fig 5 shows the organisation of object frame, 5.1 and time slots, 5.2. In the case where n is a fractional number, the transmission of the object frame is scheduled earlier than the time frame bounded by the object decoding rate, as in, 5.4. Over several object frame periods, when the fractional portion of the time slots accumulates more than one, one additional time slot, 5.5, would be allocated for the object frame marked by 5.4. The allocation of time slot marked by 5.5 is strictly based on the needs determined by module 4.8 of the block diagram in fig. 4. The time frame of the object frame is dynamically updated by the object descriptor at the rate equivalent to the arrival of the network data packet of the object data.

The calculation of an elastic delay is performed by equation 1 and 2. With the preset delay being set by higher application software or set at pre-configured time, the time slot can choose which decoding frame the object can start transmitting. The subsequent object of the same reference must be transmitted

within the object decoding time or less. Each node can have a pre-configured delay for the object stream of a specificed object reference.

The preferred implementation of the timer as mentioned in blocks 4.4 and 4.12 of fig 4 and 3.6 of fig 3 is illustrated in fig 6. This implementation of the timer is used to provide synchronisation between network components. Each time a timer instance is instantiated, a timer session is created at step 6.1. An individual timer instance provides a means to insulate the time reference from other object stream object references. The timer is updated with the timing information from the object descriptor, at step 6.2. The time value in the object descriptor provides the reference decoding time of the object data. All object references with similar major reference number and with the same source and destination address with similar network service access point numeral representative use the timer session. This allows all objects within the same object stream to follow one timing reference. Delta_time refers to the time resolution of the timer circuit. It can be used as timing information for a particular object reference stream. This value is preset based on the timing required for the object reference. Tick is a time reference that increases asymptotically based on the timer resolution adopted by the OS or some hardware clock. The Tick value is decreased by Delta_time if Tick is more than Delta_time at step 6.25. As the object data forming the same object stream arrives, the timer is updated with the internal time, delta_time, at step 6.3, with time resolution finer than the time period of one time slot computed. The timer

session is deleted if the object stream discontinues the object stream transmission, at step 5, 6.7, 6.4.

Based on the information from the system time slot divider, time slot information for each individual object stream, and the scheduling of the data object to be streamed out to another destination is performed as shown in fig. 7. Reference numeral 7.1 represents an arbitrary number of input object traffic streams entering the node to be re-transmitted to another destination network node. Information such as the stream object descriptor and the information from the output of the time slot divider that provides description of the object are multiplexed by block 7.2. A full scan for k object streams, 7.1, by the multiplexer is performed at the maximum rate of the system object frame generated by the system time slot divider, 4.5. This can be performed at a faster rate if not all the streams have objects for delivery. Besides streaming the information to the stream object slot scheduler, the multiplexer also keeps the statistic of the object stream, as shown in the data structure below. This allows feedback for the dynamic mechanism required by the traffic shaper or filter.

```
Object Stream Statistic {
        Max Object Size in Bytes;
        Min Object Size in Bytes;
        Average Decoding Rate;
        Total Objects received;
}
```

In a preferred form, the multiplexing sequence starts from the object stream with the least decoding rate of 7.1 and the other precedence classifications then follow.

Fig. 7 shows the detailed description of the object packet stream traffic control. The object stream shaping block 7.3, performs filtering based on the object data size, the rate of entry and the precedence of each of the objects received from one single stream. The detail of the shaping algorithm can vary depending on the rate of decoding. For the data that meets the filter criteria, reference to the data object is passed to stream object slot scheduler, 7.4. The stream object scheduler then assigns the object to the time slot to deliver the object within the decoding rate as specified in the object descriptor. The time slot schedule information which refers to the object delivery time, in terms of time slots, is passed to the time slot allocator. References of data objects that did not meet the filter requirements are passed to an object cache control, 7.5 where information of the cache for the object is passed to the time slot allocator's stream object cache memory of fig 8. The object streams in the cache memory are object precedences of lower level which should not be transmitted because the capacity for the stream has reached the assigned bandwidth according to the filter in the object stream shaping block 7.3.

Fig. 8 shows a functional diagram of the time slot allocator. Reference numeral 8.1 represents an arbitrary stream traffic received. The object frame created by

the time slot allocator is based on the assigned transmission bandwidth and the minimum decoding rate determined from system time slot divider, 4.3. The time slot schedule information from the object stream traffic control, 4.8, determines if the data selected by the object stream multiplexer should put the object on a stream object queue, 8.3 or a stream object cache memory, 8.5. Based on the queue information from the stream object queue and the stream object cache memory, the object is slotted into the time slot of the object frame. Slotting of object data into a time slot is performed by a object to time slot mux, 8.4. Because of its finite memory size, the stream object cache memory 8.5 may choose to discard the longest staying data objects if the memory assigned is used up. If there are free time slots after all the object in the stream object queue as been allotted, the data object in the stream object cache memory is multiplexed out via the object to time slot mux, 8.4, to be delivered out to the network. The minimum queue size for each stream should not be less than the entire object frame size. The total memory assigned for the object stream in both cache memory and queue memory should not exceed the following to meet the total group delay for one node;

Memory Size (Queue & Cache) = Group Delay X (max. Object Size / Decoding Rate) (EQ. 3)

Depending on the amount of delay to be supported, the group delay value of each node need to be updated such that re-allocation of the memory size can

be recomputed. The recomputed memory size using eq.3 determines the maximum group delay per hop.

For each object frame computed dynamically by the functional block, 4.3, for the system implementing the exchange object traffic controller, a new number of time slots per object are computed using the conditions set forth in eq.1 and eq.2. The rate of change is performed at the object frame rate. Fig 9 shows the flowchart for the allocation and de-allocation of time slots being carried out in the functional block 4.8. The flowchart assumes that the allocation of all objects based on the allotted bandwidth for the intermediate node and also the precedence criteria to be implemented accordingly for the streams to be delivered is performed based on the preference of one skilled in the art. At step 9.1, the number of excess time slots is initiated at zero. For each object frame, the number of time slots is computed by functional block, 9.2, meeting the criteria set forth in eq.1. Object data to be delivered within the object frame is determined at step 9.3 based on the data packet in the stream object queue 8.3. If the objects in both the stream object queue 8.3 and stream object cache memory 8.5, have data objects that do not meet the real-time requirement, tested at step 9.4, packets with the object reference are dropped accordingly at step 9.5. Objects that meet the real-time requirement meet the criterion that the total time the object data packets resides in the queue is less than the decoding rate of the object stream. Schedule time (t) is defined as the total time spent in the queue and time spent if the object is delivered in the next object frame. If the schedule time (t) is larger than the decoding rate of the object, this means

that the data object needs to be delivered in the current object frame. If additional time-slots are required within the object frame, a check is made at step 9.7, if there are additional time slots saved over from the previous object frame. The additional time slots re-allocated are updated at step 9.8, which keeps a record of global time slots reserved over an arbitrary period of time. With the excess time slots, data objects in the stream object cache memory are scheduled for delivery next, performed based on criteria in step 9.9. Assignment of data objects based on the allocated time slots is performed in 9.10. After all objects with the decoding time frame have been assigned, the remaining time slots for the object frame are saved for future object frames at step 9.11.

The thus described embodiment of the invention provides system architecture for the apparatus as described in the object traffic controller to be configured and connected to any legacy network structure for end-to-end delivery of real-time data. The apparatus provides a real-time mechanism for the content switching based on timing requirement of individual data streams at the network node performing the routing and switching functions. Network congestion is avoided by discarding object data which may consists of a number of network data packets if the real-time requirement failed to be observed due to heavy network traffic. Discarding objects failing real-time requirement allows efficient use of network resources and also provide an indicator to higher layer software that the bandwidth is insufficient for the session. It also enables the network components to operate in a collaborative effort in streaming real-time traffic

through a large network. It also frees up the additional resources required at the client terminals to implement sophisticated protocol stack for providing feedback based on the traffic received. In addition, the system allows real-time with high timing accuracy to be delivered across a wide area network without drastic change to the existing network architecture. The client can receive quality content with graceful degradation when there is network congestion or when the bandwidth to the client reduces due to heavy utilisation of network resources or server resources. On the service and content provider ends, it provides an infrastructure and means to differentiate the real-time services provided for the client based on client's profile. This invention is very useful in delivering real-time data across heterogeneous network to light-weight client with limited network resources and processing power such a mobile PDA or mobile video terminal.

CLAIMS

1. A method of sending a data object through a communications network from a sender to a recipient via at least one network node comprising the steps of:

 a. dividing the data object into a stream of data packets to be transmitted,

 b. marking each data packet with a delivery time; and

 c. discarding a said data packet en route when the delivery time cannot be met.

2. A method as claimed in claim 1 or claim 2 wherein the data object has associated therewith a descriptor, the descriptor providing information linking the data packets of the object.

3. A method as claimed in claim 3 wherein the descriptor further includes the decoding rate for the object.

4. A method as claimed in claim 3 wherein the descriptor further includes the delivery time information.

5. A method as claimed in claim 1 wherein step (a) includes the step of marking each packet with a first reference, common to all packets of the object and a second reference unique to that packet.

6. A method as claimed in any one of the preceding claims wherein the node calculates a delivery time slot for each packet received based on the decoding rate for the object and the transmission time required.

7. A method as claimed in claim 6 wherein the node determines a priority for the sending of the packet onward through the network in accordance with the delivery time slot.

8. A method as claimed in claim 7 wherein each data packet is marked with a precedence level, the node determining the priority in accordance with the precedence level as well as the delivery time slot.

9. A method as claimed in claim 7 or claim 8 wherein the node stores lower priority packets for future sending.

10. A method of creating synchronised time stamped multimedia objects comprising the steps of:

a. fragmenting the data objects into one or more data packets;

b. marking each data packet containing a part or full portion of a single multimedia object using a unique reference;

c. creating a link for related fragmented network data packets originating from the same multimedia object using an object reference before transmission;

d. stamping the data packets related to the object reference with a time reference marking the instance the data packet leaves a transmission node; and

e. attaching a precedence level of the object based on the object reference.

11. A method related to claim 10, further comprising the step of providing an object descriptor for each multimedia object to allow a means for higher entitiy software to specify a data object linking mechanism and to synchronise multimedia objects delivered across network in a form of fragmented network data packets.

12. A method for scheduling the transmission time of multimedia objects comprising the steps of;

a. dividing the transmission time into frames based on an object decoding rate;

b. further dividing the transmission time within each frame of the multimedia object channel into time slots, the number of time slots being equivalent to dividing the transmission rate by the multimedia object decoding time;

c. forming a delivery time schedule of multimedia objects with a time stamp within the scheduled object delivery period or frame period;

d. computing a scheduled object delivery period which is equivalent to the maximum object group delay period allowable; and

e. re-scheduling objects that cannot be delivered within a current time slot to a later time slot.

13. A method as claimed in Claim 12, further comprising the step of dynamically changing the number of time slots per frame by adapting to a changing object decoding rate of the object descriptor and changes to the transmission speed of network link.

14. A method as claimed in claim 12 or claim 13, comprising the further steps of computing, allocating and de-allocating the number of time slots dynamically based on current frame by:

a. computing the number of time slots available for multimedia object delivery in the current object frame;

b. updating the transmission time slot reserved over buffered object data in both queue memory and cache memory of a stream connection;

c. de-allocating the time slots of a subsequent frame if transmission of intended multimedia objects cannot be delivered within the assigned frame period; and

d. allocating time slots accumulated from previous object frames if transmission of intended multimedia objects within the

specified decoding time frame cannot be delivered within the assigned object frame.

15. A method according the Claim 12, further comprising the step of computing a queue buffer size required for a stream based on the maximum object group delay and maximum multimedia object size.

16. A method as claimed in claim 12, further comprising the step of calculating an elastic delay period for the arrival of multimedia objects based on the maximum object group delay, consisting of at least one of the following delays:

a. end to end delivery delay time for all objects at a value equivalent to the maximum object delay.

b. node re-transmission delay where the maximum object delay is at a value not more than the n multiples of the maximum object delay experienced at each intermediate node between sender and receiver; and

c. a queue buffer delay introduced by the queuing algorithm at each intermediate nodes where the maximum delay is based on the size of the queue buffer size divided by the minimum object size.

17. Apparatus for interconnecting an object streaming apparatus into existing network interconnection apparatus to control data traffic generated by the multimedia object streams comprising:

a. an ingress gateway to control entry of data packets into a network;

b. a content traffic exchange hop attached to legacy switching or a routing network component; and

c. an egress gateway to control the exit of data packets from the network and to stream content to end-users.

18. Apparatus as claimed in claim 17 wherein the ingress gateway comprises:

a. an object fragmentor to break up an object into smaller data packets suited for the payload size of the network delivery medium;

b. a timer used for performing a clocking function to emulate the decoding time of the object to be decoded by the end device;

c. an object marker to provides linking of network data payloads belonging to the same data object to be delivered;

d. a time slot divider to compute the transmission time of the object data fragmented in a number of data packets;

e. a time slot allocator to determine the exact time of delivery; and

f. a network packet formatting means.

19. Apparatus as claimed in claim 17 or claim 18 wherein the content traffic exchange hop and egress gateway comprise at least one of:

a. a network parameter mapper and decoder module arranged to check and encode headers of packetised multimedia objects , each header including a precedence level, a time stamp of the multimedia object and the source and destination reference;

b. a network payload packetizer and de-packetizer arranged to encapsulate and assemble the object data from the data packets of same object received from and transmitted to the network layer;

c. an object stream scheduler arranged to register the decoding time for each data object stream received and to schedule the delivery time of the stream to be routed out of the node;

d. a data packet transmission scheduler arranged to schedule the transmission of a single multimedia object if transmission time allows the single multimedia object to be sent out in entirety;

e. a time slot re-scheduler arranged to re-allocate the sending time of objects to an earlier time slot if the objects failed to meet the real time requirement of the object stream;

f. a system time slot divider arranged to synchronise all object streams for re-transmission or re-routing; and

g. a time slot allocator to arrange to provide an exact delivery time of object streams leaving the node;

20. Apparatus as claimed in any one of claims 17 to 19 further comprising a clocking mechanism at each host and routing node to synchronise with the multimedia object streams received and to schedule the transmission time of object streams at determined time slots.

21. Apparatus as claimed in claim 20 wherein the mechanism comprises:

a. a module arranged to determine the arrival of the first data packet belonging to an object based on a multimedia object reference;

b. a module arranged to compute and maintain a time table of all the streams based on the difference between the last sent object and the next scheduled transmission of object;

c. a module arranged to create a link list for the data packets received for all subsequent data packets belonging to the same object; and

d. a queuing module to put data packets in assigned time slots.

22. Apparatus as claimed in any one of claims 17 to 21 further comprising means of avoiding network congestion and limiting short-term bandwidth utilisation peaks by delaying the transmission of data objects by using cache memory for data objects to meet real-time decoding at the end receiver.

23. Apparatus as claimed in Claim 22, further comprising means of providing synchronisation of object streams.

24. Apparatus as claimed in any one of claims 17 to 23 further comprising means for moderating peak data by diverting object data of lower precedence to a cache memory .

25. Apparatus as claimed in any one of claims 17 to 24 further comprising means for enabling a different traffic filter specification based on a generic method for allocation of bandwidth by dividing a real-time delivery window in terms of decoding frames into multiple time slots.

ABSTRACT

TIME BASED MULTIMEDIA OBJECTS STREAMING APPARATUS AND METHOD

A method of and apparatus for sending a data object through a communications network from a sender to a recipient via at least one network node, the method including the steps of: (a) dividing the data object into a stream of data packets to be transmitted, (b) marking each data packet with a delivery time; and (c) discarding a said data packet en route when the delivery time of the said packet cannot be met.

Figure 1.

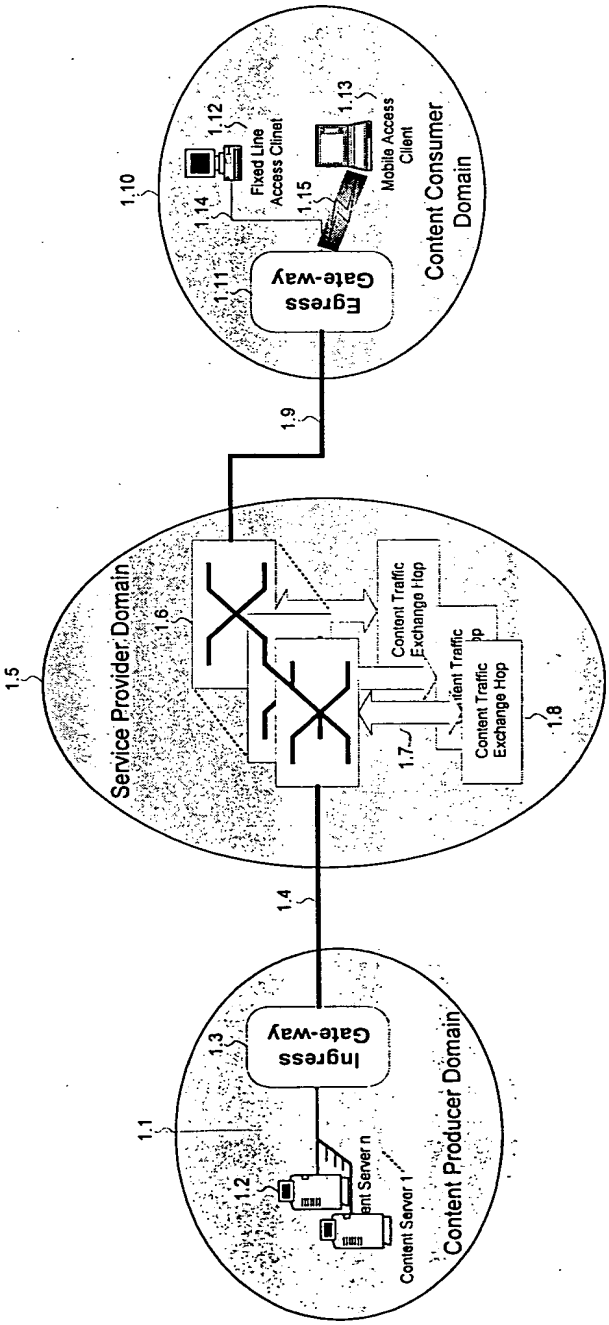

Content Producer Domain
Ingress Gate-way
1.1
1.2
1.3
Content Server 1
Content Server n
1.4
Service Provider Domain
1.5
1.6
Content Traffic Exchange Hop
Content Traffic Exchange Hop
1.7
1.8
1.9
Content Consumer Domain
1.10
Egress Gate-way
1.11
Fixed Line Access Clinet
1.12
Mobile Access Client
1.13
1.14
1.15

FIG. 1

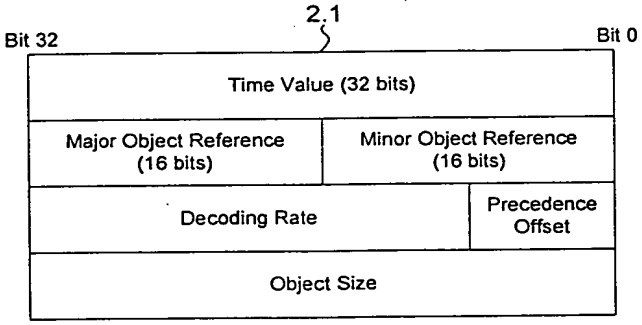

2.1
Bit 32
Bit 0
Time Value (32 bits)
Major Object Reference (16 bits)
Minor Object Reference (16 bits)
Decoding Rate
Precedence Offset
Object Size

FIG. 2a

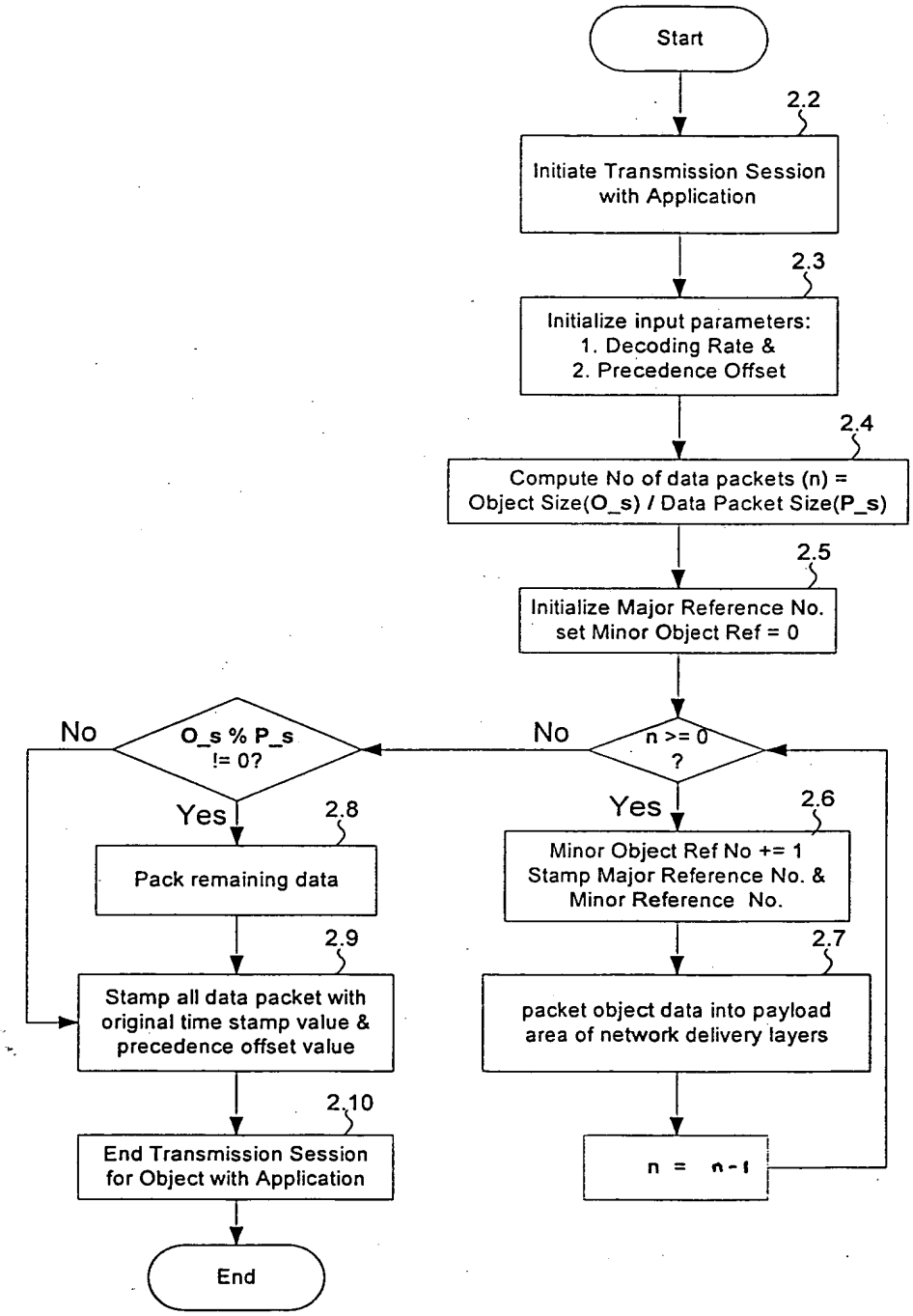

Start
2.2
Initiate Transmission Session with Application
2.3
Initialize input parameters: 1. Decoding Rate & 2. Precedence Offset
2.4
Compute No of data packets (n) = Object Size(O_s) / Data Packet Size(P_s)
2.5
Initialize Major Reference No. set Minor Object Ref = 0
n >= 0 ?
No
Yes
2.6
Minor Object Ref No += 1 Stamp Major Reference No. & Minor Reference No.
2.7
packet object data into payload area of network delivery layers
n = n-1
O_s % P_s != 0?
No
Yes
2.8
Pack remaining data
2.9
Stamp all data packet with original time stamp value & precedence offset value
2.10
End Transmission Session for Object with Application
End

FIG. 2b

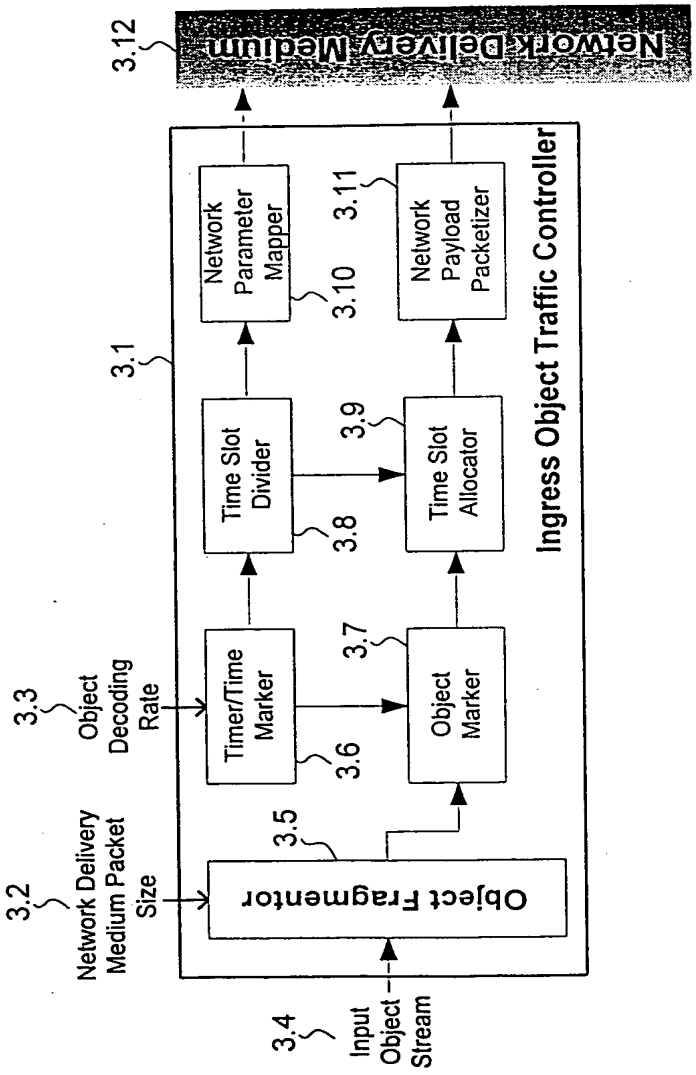

Network Delivery Medium
3.12
Ingress Object Traffic Controller
3.1
Network Parameter Mapper
3.10
Network Payload Packetizer
3.11
Time Slot Divider
3.8
Time Slot Allocator
3.9
Timer/Time Marker
3.6
Object Marker
3.7
Object Decoding Rate
3.3
Network Delivery Medium Packet Size
3.2
Object Fragmentor
3.5
Input Object Stream
3.4

FIG. 3

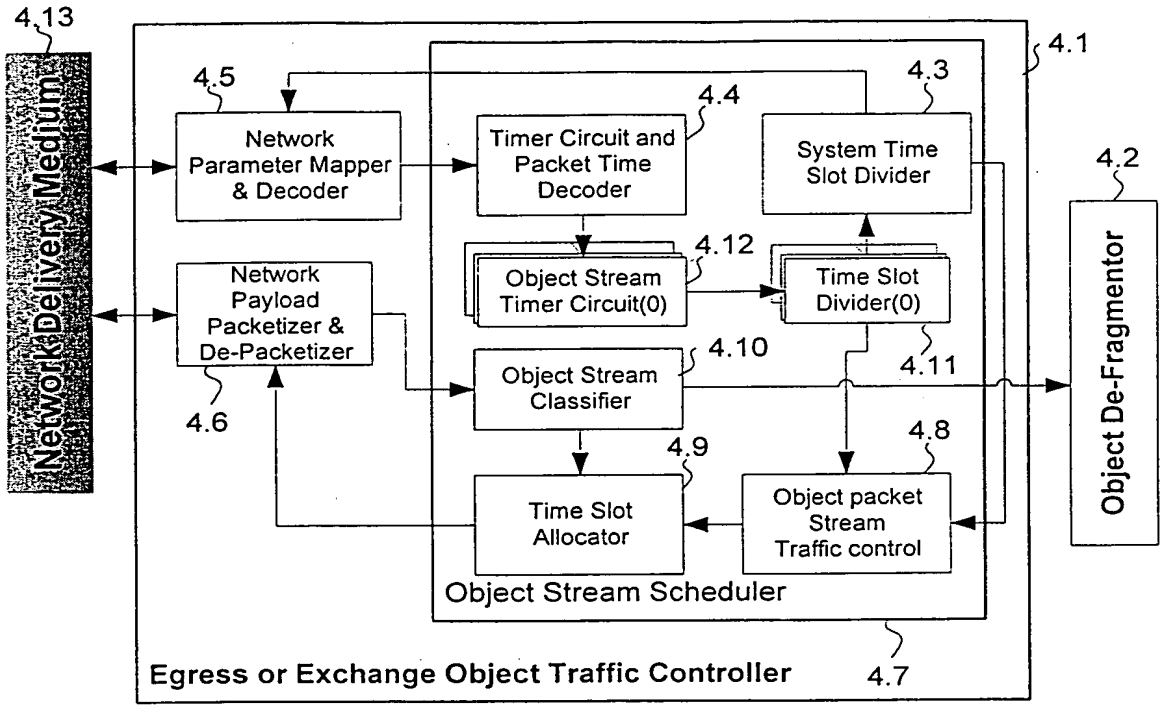

4.13
Network Delivery Medium
4.5
Network Parameter Mapper & Decoder
4.4
Timer Circuit and Packet Time Decoder
4.3
System Time Slot Divider
4.1
4.2
Object De-Fragmentor
4.12
Object Stream Timer Circuit(0)
Time Slot Divider(0)
Network Payload Packetizer & De-Packetizer
4.6
4.10
Object Stream Classifier
4.11
4.9
4.8
Time Slot Allocator
Object packet Stream Traffic control
Object Stream Scheduler
Egress or Exchange Object Traffic Controller
4.7

FIG. 4

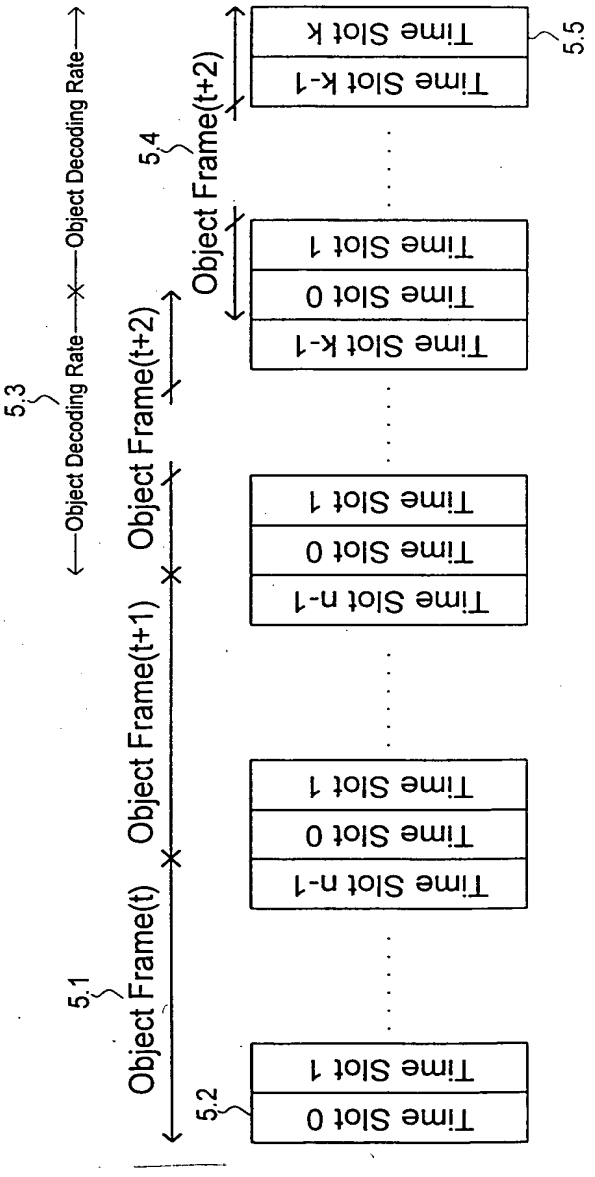

Object Frame(t)
Object Frame(t+1)
Object Frame(t+2)
Object Frame(t+2)
Time Slot 0
Time Slot 1
Time Slot n-1
Time Slot k-1
Time Slot k
Object Decoding Rate
Object Decoding Rate
5.1
5.2
5.3
5.4
5.5

FIG. 5

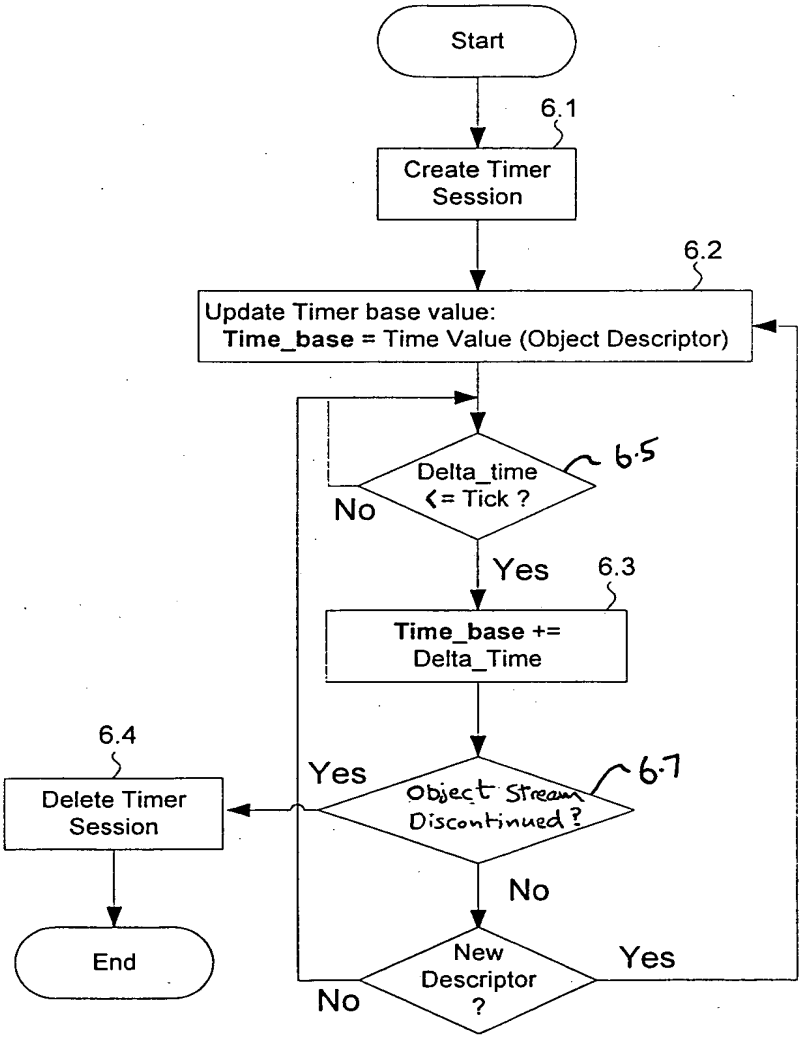

Start
6.1
Create Timer Session
6.2
Update Timer base value:
Time_base = Time Value (Object Descriptor)
Delta_time <= Tick ?
6.5
No
Yes
6.3
Time_base += Delta_Time
6.4
Delete Timer Session
Yes
Object Stream Discontinued ?
6.7
No
End
New Descriptor ?
Yes
No

FIG. 6

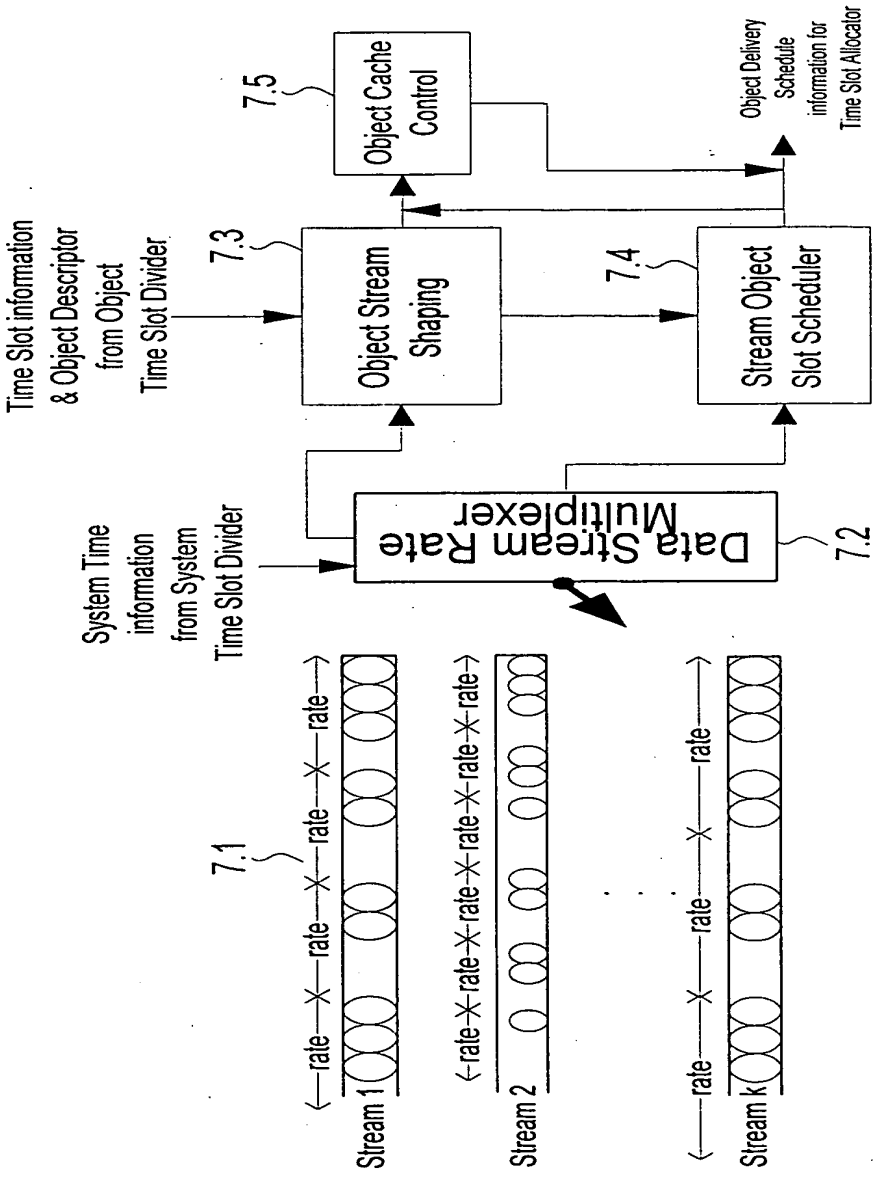
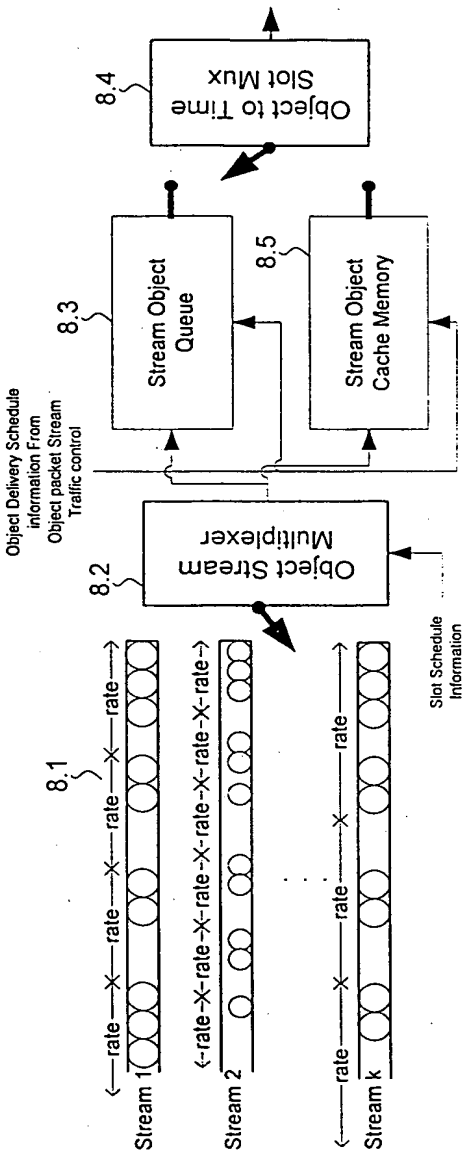

Object Cache Control
7.5
Object Delivery Schedule information for Time Slot Allocator
Time Slot information & Object Descriptor from Object Time Slot Divider
7.3
Object Stream Shaping
7.4
Stream Object Slot Scheduler
System Time information from System Time Slot Divider
Data Stream Rate Multiplexer
7.2
7.1
rate
Stream 1
Stream 2
Stream k

FIG. 7

FIG. 8


Object to Time Slot Mux
8.4
Stream Object Queue
8.3
Stream Object Cache Memory
8.5
Object Delivery Schedule information From Object packet Stream Traffic control
Object Stream Multiplexer
8.2
Slot Schedule Information
8.1
Stream 1
Stream 2
Stream k
rate

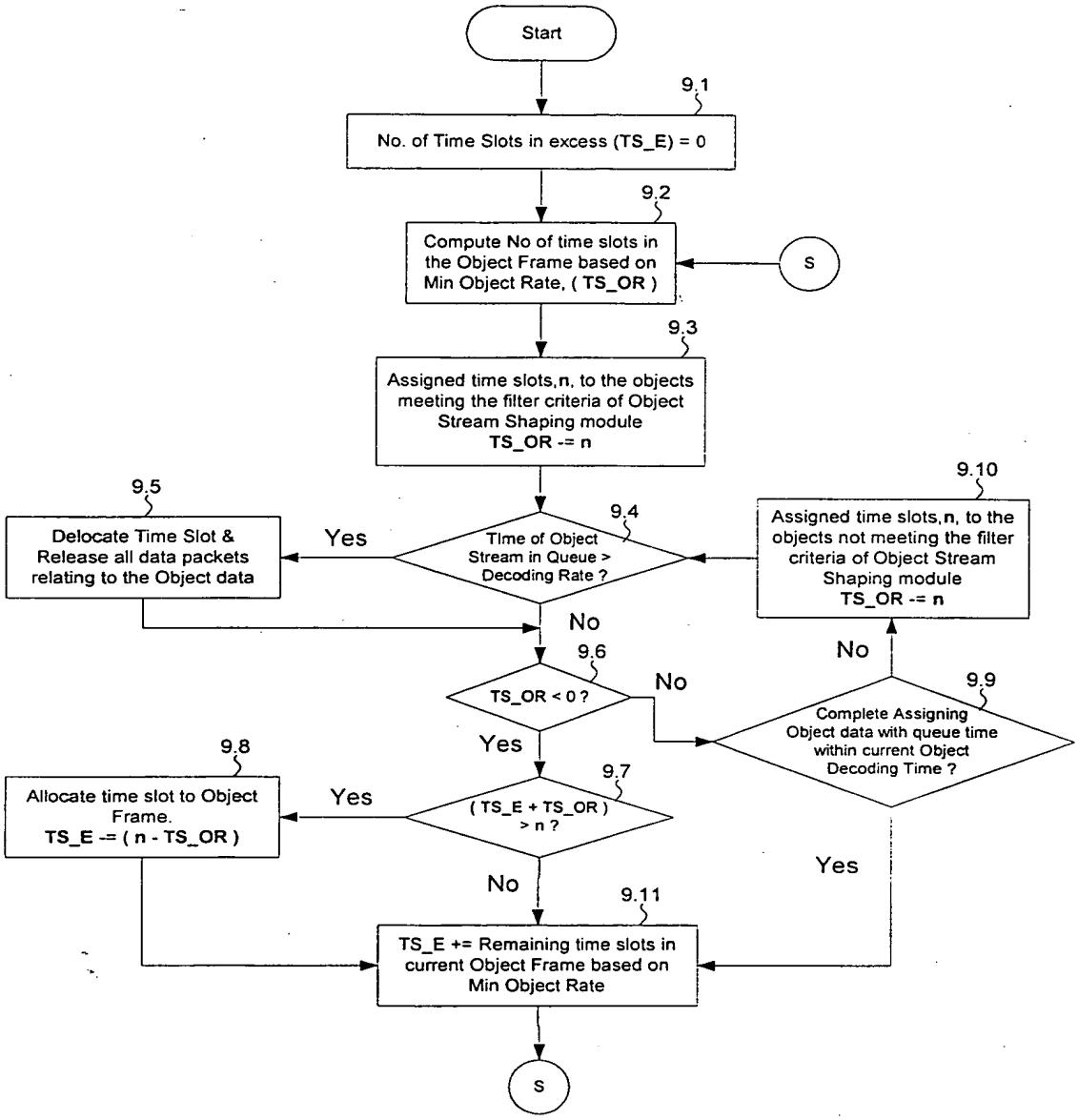

Start
9.1
No. of Time Slots in excess (TS_E) = 0
9.2
Compute No of time slots in the Object Frame based on Min Object Rate, (TS_OR)
S
9.3
Assigned time slots,n, to the objects meeting the filter criteria of Object Stream Shaping module
TS_OR -= n
9.5
Delocate Time Slot & Release all data packets relating to the Object data
Yes
9.4
TIme of Object Stream in Queue > Decoding Rate ?
9.10
Assigned time slots,n, to the objects not meeting the filter criteria of Object Stream Shaping module
TS_OR -= n
No
9.6
TS_OR < 0 ?
No
9.9
Complete Assigning Object data with queue time within current Object Decoding Time ?
Yes
9.8
Allocate time slot to Object Frame.
TS_E -= (n - TS_OR)
Yes
9.7
(TS_E + TS_OR) > n ?
No
Yes
9.11
TS_E += Remaining time slots in current Object Frame based on Min Object Rate
S

FIG. 9